

Independent Assessment of Central Maine  
Power Management Practices for  
SmartCare Implementation

# Testimony

## Independent Assessment Report

Docket: 2019-00015

**Presented to:**

State of Maine Office of the Public Advocate



**Submitted by:**

Laurel Arnold, Senior Manager  
BerryDunn  
100 Middle Street  
Portland, ME 04104  
[larnold@berrydunn.com](mailto:larnold@berrydunn.com)

**Submitted on:**

September 6, 2019

---

## Table of Contents

| Section                                   | Page |
|---|------|
| Table of Contents.....                    | i    |
| 1 Introduction .....                      | 1    |
| 1.1 Assessment Scope and Objectives ..... | 1    |
| 1.2 Qualifications .....                  | 1    |
| 1.3 Background and Methods .....          | 2    |
| 1.4 Summary of Key Findings .....         | 4    |
| 2 Findings .....                          | 5    |
| 3 Conclusions .....                       | 30   |
| 4 Recommendations .....                   | 31   |
| Attachment 1: Laurel Arnold Resume.....   | 33   |

## 1 Introduction

### 1.1 Assessment Scope and Objectives

The Maine Office of the Public Advocate (OPA) retained BerryDunn to determine whether Central Maine Power Company (CMP or the Company) used sound management practices to implement the SmartCare system. I was assigned to conduct this review. I was assisted in this effort by Bill Brown, Julie Keim and Leila Hanna.

### 1.2 Qualifications

My experience and expertise make me well qualified to assess the practices used on the SmartCare project. I have overseen and assessed a wide range of internal and consumer facing technology implementation projects with budgets ranging from less than \$100K up to \$500M. My resume, provided as “Arnold Attachment 1,” highlights my experience. I often advise clients on high-risk and at-risk projects, project recovery efforts, and development of practices to minimize project risk and increase the probability of project success. The following are some relevant examples:

- **State of West Virginia, Bureau for Medical Services (2003-2012)** – During my 8 years advising the Bureau for Medical Services, I provided independent quality assurance and project oversight. My work on project recovery and system re-implementation for this client is especially relevant.

Shortly after the initial implementation of its pharmacy claims system, the West Virginia Bureau for Medical Services invoked its rollback contingency plan, reverting to its prior system for processing of claims. I was asked to provide independent project oversight for the project recovery and system remediation. I worked with the State and Fiscal Agent teams to rebuild team morale, identify the root causes of the initial failure, validate remediation of defects and objectively verify readiness for re-implementation. The re-implementation was deemed a success by the Bureau for Medical Services and the Pharmacy Point of Sale system was subsequently certified by the Center for Medicare and Medicaid Services.

Once the system had been remediated, I assisted the Bureau in identifying suspect transactions processed by the system prior to rollback and helped develop a plan to address over and under payments through reprocessing of transactions and adjustments. As an objective third party, I helped the Bureau and their vendor rebuild stakeholder confidence and reach resolution on over and under payments to providers.

I subsequently helped the Bureau for Medical Services to strengthen strategic planning, portfolio management and project management capabilities to help realize program goals and objectives.

- **State of North Carolina, Office of State Auditor (2007)** – I led an independent assessment of the Enterprise Project Management Office for the State of North Carolina. The Office of the State Auditor (OSA) engaged BerryDunn to provide an independent evaluation of the standards and accountability measures created and monitored by the State of North Carolina’s Information Technology Services (ITS) to determine whether the policies, procedures, and practices are significantly improving the likelihood that a given project would be brought in on time and on budget. In addition, the OSA sought to determine whether IT projects, policies, procedures, and practices were being implemented in accordance with industry best practices.
- **Massachusetts Health Insurance Exchange and Integrated Eligibility System (2012-Present)** – For nearly seven years I have led the BerryDunn team, providing the Massachusetts Executive Offices of Health and Human Services and the Commonwealth Connector Authority with Independent Verification & Validation (IV&V) services for implementation of the Massachusetts Health Insurance Exchange and Integrated Eligibility System (MA HIX/IES). The team reviews both the project management and system delivery aspects of the project as conducted by the Commonwealth and its contracted vendors. We conduct deliverable reviews; verification and validation of automated code review and continuous integration results; cost allocation and financial status reports; review of expected and delivered reusability; independent assessment of implementation readiness; issue and risk management.

### 1.3 Background and Methods

On October 30, 2017, CMP implemented a new metering and billing system it calls SmartCare. This system replaced a legacy system. The primary vendor was SAP. In its final report dated December 20, 2018, the Liberty Consulting Group (Liberty) described the new system and discussed the problems it found with CMP’s implementation. Liberty subsequently published an addendum reaffirming its findings. The projected project cost was more than \$57 million and required nearly two years to complete. It involved more than 240 resources, 32 vendors in 6 geographical locations and required interaction with 52 interfaces.

Many project activities and responsibilities for the execution and system implementation in the SmartCare project were outsourced. This testimony focuses on the subset of processes and practices that were primarily the responsibility of CMP as the sponsor and entity ultimately in charge of oversight and decision making.

I examined initial planning decisions as well as the continuing action of the Company in response to changing circumstances and in light of facts and circumstances that either were known or knowable at the time decisions were made. I then compared CMP’s management of the project to widely accepted management and testing practices in the industry, leading me to the findings and recommendations contained herein.

The body of good management practices of a project such as the SmartCare project has been well documented for more than two decades. The Project Management Body of Knowledge (PMBOK®)<sup>1</sup> is comprised of two parts – the Guide to the Project Management Body of Knowledge and the Standard for Project Management. The second part of the PMBOK®, the Standard for Project Management, is part of an American National Standard (ANSI) developed in accordance with ANSI's requirements including public review and comment and a consensus process.

The PMBOK® describes good practices as those where the application of the knowledge, skills, tools, and techniques can enhance the likelihood of success in the delivery of a project. The PMBOK®'s recommended practices are applicable to most projects most of the time. It provides guidance for tailoring project management processes and how tools and techniques can be applied to projects. I relied upon the PMBOK® standards and my experience in project management to reach the conclusions contained in this testimony.

Software Development Life Cycles (SDLCs) methodologies, which define the process of developing an information system with proper analysis, design, implementation, and maintenance were continuously evolving and maturing from the mid-1970s to the early 2000s. By the mid-2000s, the great body of knowledge and limitless real-world case studies provided opportunities for research and scrutinizing of processes, approaches, and techniques. Industry standards were refined and became a set of obligatory requirements established by a general agreement and currently maintained by a recognized body to impose a uniform disciplined practice. Standards provide the guidelines for industry best practices. Best practices are a continuously evolving process of comparing business performance metrics and processes with the best-known methods of the industry.

The primary sources for software industry standards are:

- The Institute of Electrical and Electronics Engineers (IEEE) publishes tutorials and standards related to electrical and electronics engineering and computer science fields;
  - The International Organization for Standardization (ISO) has developed and published more than twenty thousand standards covering everything from technology, software quality, manufactured products, agriculture, and healthcare;
  - The Capability Maturity Model (CMM), funded by the U.S. Department of Defense, is a development model created by the software community in 1986; and
  - The International Software Testing Qualifications Board (ISTQB), founded in 2002, is a software testing certification board that operates internationally.
- 

<sup>1</sup> Project Management Body of Knowledge (pmi.org) originally published in 1996, is now in its 6<sup>th</sup> edition.

From the standards established by these organizations, best practices continue to evolve and are published within research journals, white papers, tutorials, lectures, and online blogs. These sources and my experience in successful large-scale software applications design, development, verification and validation, and production release informed me in reaching my conclusions regarding testing and defect management.

This testimony presents the results of my assessment activities, which included:

- Review of the responses that CMP provided to the data requests issued by The Liberty Consulting Group (Liberty) during the course of its audit of CMP.
- Review of Liberty's Final Report Forensic Audit of CMP's Metering and Billing Systems (December 20, 2018).
- Review of answers and materials provided by CMP in response to OPA-007.
- Review of testimony from CMP employees provided during the technical conference conducted April 29, 2019 in the companion case, Docket No. 2018-00194.
- Review of CMP responses to ODR-011 from the April 29, 2019, technical conference in Docket No. 2018-00194.
- Review of materials provide in response to OPA-008 in this docket.
- Review of information provided during the technical conference conducted June 16, 2019 in this case.
- Review of responses to ODR-003 from the June 16, 2019, Technical Conference in this docket.
- Review of responses to TLCG-001.

#### **1.4 Summary of Key Findings**

- Based upon review of the information provided, I concur with conclusions in Section VII of the Liberty Report relative to the management of the SmartCare project.
- CMP management practices for testing, defect management and requirements did not align with best practices. CMP management did not develop and use a complete and objective status of system readiness prior to Go-Live.
- CMP did not manage project risk consistently and effectively.

## 2 Findings

In conducting my work, I made every effort to review data in a logical sequence and within the context of the associated phase of the SmartCare project to ascertain if practices and decisions were reasonable in light of existing facts and circumstances that either were known or knowable at the time decisions were made.

### **Finding 1: I concur with Liberty Report's assessment and conclusions in Section VII Customer Information System Implementation relative to the management of the SmartCare project.**

The Maine Public Utilities Commission (Commission or MPUC) retained Liberty to conduct a forensic audit of CMP's metering, billing, and related systems in the wake of a period of high electricity usage registration and large numbers of customer complaints and inquiries beginning in November 2017.

In the course of examining the impact the SmartCare (also known as Customer Information System (CIS)) implementation and post Go-Live operation had on billing and customer interaction in November 2017 through April 2018, the Liberty Group submitted data requests to CMP, the responses to which are relevant to my assessment.

Based upon my review of responses provided to Liberty's data requests and my review of Liberty's Final Report, I concur with the Liberty Group's conclusions relative to practices used for the SmartCare project. More specifically:

- The detailed project schedule created by the Project Management Office was not effectively used to manage integration of effort, understand the downstream implications of specific delays or effectively mitigate risk.<sup>2</sup>
- Inconsistent practices in status reporting negatively impacted efforts to identify and resolve problems prior to Go-Live. Conflicting information in status reports, readiness reports, and the master list of defects demonstrate a lack of recognition of the magnitude of the existing issues.<sup>3</sup>
- Management's decision to allocate a fixed time period or "time box" for the completion of testing to meet the October 30, 2017 Go-Live date required a significant modification to

---

<sup>2</sup> Liberty Final Report p. 67-68.

<sup>3</sup> Liberty Final Report p. 65-67.

the testing approach. The results of the change in testing approach introduced risk, increasing the likelihood that defects would remain undetected until after Go-Live.<sup>4</sup>

**Finding 2: CMP's pre-implementation testing did not align with best practice or the approved strategy.**

Deloitte, hired by CMP as the SmartCare System Integrator, prepared a Testing Strategy for CMP that was approved by CMP. CMP provided this document as Attachment 1 in response to OPA-008-011. Section 1.1 states that the Testing Strategy document will cover details of “the types of testing, testing tool use, standard best practices, test documentation standards, defect resolution and escalation processes, and potential testing risks along with mitigating factors.” My colleague Leila and I reviewed this document carefully and found that the Testing Strategy did not align with best practice. In this section of the report we will outline the ways in which CMP's adopted Testing Strategy failed to meet industry standards and best practices. Furthermore, CMP did not follow the Test Strategy that they approved.

***The list of test types was incomplete and the description of each type deviated from best practice.*** The purpose of testing as described in Section 1 at page 4 of the Testing Strategy is to help ensure that the implemented system has met requirements for the project.

The Testing Strategy document set forth the following types of testing:

- Unit Testing
- String Testing
- Integration Testing
- User Acceptance Testing
- Performance and Stress Testing
- Regression Testing
- Parallel Testing for Billing
- Converted Data Testing

This list of testing types is incomplete and does not include all of the types of testing required by best practices for large software development and implementation efforts. Without completing

---

<sup>4</sup> Liberty Final Report p. 60-63.



all types of testing, CMP could not know all the possible issues and defects that remained when the system was implemented. Testing types within the test phases of the Software Test Life Cycle (STLC) should include system testing and security testing phases, as well as smoke testing, also known as confidence testing or post-deployment testing.

System testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.<sup>5</sup> System testing occurs after integration testing is complete.

Smoke testing is an important precursor to all test phases and a necessary element of Cutover Testing. Smoke tests are a subset of all defined, planned test cases that cover the main functionality of a system to ascertain that the most crucial functions of a program work.<sup>6</sup> Smoke testing is executed after each code delivery (aka code drop) to help ensure the software is ready and stable for testing or production release.

Within the list of testing types in the testing strategy, the descriptions provided for several of the testing types did not align with industry standards or best practices. For example:

- *Integration Testing* – The purpose of integration testing, as stated in the Testing Strategy approved by CMP, did not follow industry standards. In Section 2.3, on page 8, “Integration Testing” states: “Cycle 3 testing will include testing with external systems.” Per industry standards, integration testing is “testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them.”<sup>7</sup> Integration testing is component testing; it is performed to expose defects in the interfaces and in the interactions between integrated components. Integration testing does not include testing external systems. Testing the integration of systems and packages, including interfaces to external organizations, is conducted during system testing.<sup>8</sup>

---

<sup>5</sup> The Institute of Electrical and Electronics Engineers. (1990). *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12 [1990 Edition]).

<sup>6</sup> International Software Testing Qualifications Board (ISTQB). *Standard Glossary of Terms used in Software Testing*, Version 3.2.

<sup>7</sup> The Institute of Electrical and Electronics Engineers. (1990). *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12 [1990 Edition]).

<sup>8</sup> International Software Testing Qualifications Board (ISTQB). *Standard Glossary of Terms used in Software Testing*, Version 3.2.

System testing should not be merged into the final cycle of integration testing.<sup>9</sup> System testing should occur after interface testing of components is complete and defects related to component interfaces are classified. New defects detected during system testing can then be reliably classified as related to system interfaces. Without a clean break between integration and system testing, the origin and root cause of defects is more difficult to identify. This leads to a risk that defect analysis is more time consuming, and resolution is ineffective or incomplete.

- *User Acceptance Testing (UAT)* – In Section 2.4, on page 8, the Testing Strategy states that “UAT may be a separate test or could occur as part of the last cycle of Integration Test.” Best practices<sup>10 11 12</sup> supported by industry-standard definitions<sup>13</sup> establish UAT as the test phase in which business users validate that the specified requirements have been met by the delivered product. In most system development lifecycle methodologies, business users should not be given a version of the software for their testing until Quality Assurance has successfully completed all testing. Starting acceptance testing before the completion of integration testing risks distributing a buggy and immature system to the business users, duplicated logging of bugs by business users and testers, and expanded turnaround time for triage of the duplicate defect entries.
- *Security Unit Testing* – In Section 2.1, on page 7, under Unit Testing, the Testing Strategy references security testing as Security Unit Testing, and explains that “Individual business roles are tested to ensure they fulfill the business role designs.” Per industry standards and best practices, this approach would not provide adequate security testing.<sup>14</sup> The accepted industry definition of software security is the degree to

---

<sup>9</sup> Mochal, Tom. (2001). Integration testing will show you how well your modules get along. *TechRepublic*. Retrieved 12 August 2019, from: <https://www.techrepublic.com/article/integration-testing-will-show-you-how-well-your-modules-get-along>.

<sup>10</sup> Dias, Francis. (2016). *UAT vs. SIT Testing in QA*. Retrieved 12 August 2019, from tCognition: <https://www.tcognition.com/blog/uat-vs-sit-testing-in-qa>.

<sup>11</sup> Peham, Thomas. (2018). *Let's UAT: A practical test case illustrated on the example of Trello*. Retrieved 12 March 2019, from Usersnap: <https://usersnap.com/blog/user-acceptance-testing-example>

<sup>12</sup> Pardeshi, Shruti N. (2013). Study of Testing Strategies and Available Tools. *International Journal of Scientific and Research Publications*, 3(3).

<sup>13</sup> The Institute of Electrical and Electronics Engineers. (1993). *IEEE Guide for Software Verification and Validation Plans* (IEEE Std 1059-1993).

<sup>14</sup> Scarfone, K., Souppaya, M., Cody, A., Orebaugh, A. (2008). *Technical Guide to Information Security Testing and Assessment*. National Institute of Standards and Technology (NIST Special Publication 800-115).

which a component or system protects information and data so that persons or other components or systems have the degree of access appropriate to their types and levels of authorization.<sup>15</sup>

The testing strategy should have documented a security assessment methodology and testing objectives and strategies to provide consistency and structure to security testing to minimize risks. Security testing strategies should have included techniques for penetration tests, which simulate an attack by a malicious party, as well as tests that target the vulnerability of the system functionality related to authentication, authorization, confidentiality, integrity, and availability. The scope of security testing should have included identifying vulnerability trends, diagnosing root cause of vulnerabilities, and helping create efficient incident response procedures and remediation plans.<sup>16</sup>

Failure to define and document security testing increases the risk that vulnerabilities remain undetected. The system may be open to attacks and data breach, risking exposure of confidential customer information, or corruption of system files that could leave the system inoperable.

- *Performance Testing* – In Section 2.5, on page 9 of the Testing Strategy, Performance Testing is defined as tests that “ensure the system will perform adequately in production.” Best practices and industry definitions state that the goal of performance testing is to identify and resolve weaknesses and bottlenecks in the application.<sup>17 18</sup> Limiting performance testing to verifying adequate performance, as CMP did, precludes the primary goal of performance testing: to optimize the application for speed, capacity, and robustness, under a variety of load conditions to help ensure acceptable usability.<sup>19</sup>
- *Stress Testing* – The Testing Strategy did not fully or accurately describe the goals for stress testing or the techniques and tools that would be used. Per industry standard

---

<sup>15</sup> International Organization for Standardization. (2015). *System and Software Quality Models* (ISO/ICE Std 25010:2011).

<sup>16</sup> IBM Security. (2017). *Preempt attacks with programmatic and active testing* [White Paper]. Retrieved 20 August 2019, from IBM Corporation: <https://www.ibm.com/downloads/cas/AXDPXD6Z>.

<sup>17</sup> International Software Testing Qualifications Board (ISTQB). *Standard Glossary of Terms used in Software Testing*, Version 3.2.

<sup>18</sup> Pan, Jiantao. (1991). Software Testing. *Proceedings, International Test Conference 1991*, p.1110. DOI:10.1109/TEST.1991.519785.

<sup>19</sup> Barber, Scott. (2003). Introduction to Performance Testing. *PrefTestPlus, Inc.* Presented at PSQT/PSTT Conference Washington, DC, May 2003.

definitions, stress testing, or load testing, verifies robustness of a system.<sup>20</sup> The robustness of a software component is the degree to which it can function correctly in the presence of exceptional inputs or stressful environmental conditions. Stress tests exercise the software application within or beyond specified limits through various tools and approaches such as resource exhaustion, bursts of activities, and sustained high loads. The goal is to help ensure the software is dependable, reliable, and does not fail in unexpected or catastrophic ways.<sup>21</sup> By documenting the plan for stress testing, including techniques, tools, objectives and benchmarks, test cases can be written that reliably and accurately assess the robustness of the system and measure improvements.

The vague and inadequate definition of stress testing in the Testing Strategy results in ambiguous and incomplete testing goals. Without clear, documented, and approved testing goals, there is a risk that planned testing will be executed to completion and fail to find faults in the code. This risk increases when test cases are developed for automated load testing. Without documented objectives and intended end results, most developers will focus on creating test cases that follow successful logic paths as opposed to negative or error paths.<sup>22</sup> When a system is not fully exercised during the test phases, latent defects in areas missed during testing will be uncovered by real-world users after production release.

CMP missed an opportunity to use stress testing as a method to accelerate the rate of finding defects.<sup>23</sup> Stress or load testing simulates real-user scenarios when hundreds or thousands of users visit the application in real time. Through analysis of areas of the code base expected to be used most frequently, test analysts can focus load testing efforts on these portions of the code. Overloading, or stressing, these areas of functionality with real-world scenarios will result in the biggest return of quality. Defects

---

<sup>20</sup> The Institute of Electrical and Electronics Engineers. (1990). *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12 [1990 Edition]).

<sup>21</sup> Pan, Jiantao. (1991). Software Testing. *Proceedings, International Test Conference 1991*, p.1110. DOI:10.1109/TEST.1991.519785.

<sup>22</sup> Chambers, Kirk. (2015). *QA tips and tricks: Why a clear and robust test plan is essential*. Retrieved from Wunderman Thompson: <https://wundermanthompsonmobile.com/2015/12/qa-tips-and-tricks-why-a-clear-and-robust-test-plan-is-essential>.

<sup>23</sup> Khan, Mohd Ehmer. (2010). Different Forms of Software Testing Techniques for Finding Errors. *IJCSI/International Journal of Computer Science Issues*, 7(3), No. 1, pp. 11-16.

will be found more rapidly during stress testing than other areas of functional testing due to simulating real-world use.<sup>24</sup>

- *Cutover Testing* – Cutover Testing in Section 2.7, on page 10 of the Testing Strategy does not include a description of smoke testing, also known as post-deployment testing or confidence testing, a necessary element of Cutover Testing. Smoke testing validates the migration of business data and operations from the legacy system to the new SAP system in production. Smoke tests are a subset of user acceptance test cases, which are executed against a newly deployed application to determine if the main functions are working correctly. Smoke testing provides input to key performance indicators (KPIs) or metrics that are used to confirm all systems are operating to acceptable standards. Without smoke testing, the state of the application cannot be monitored in a reliable, reproducible, and automated manner.<sup>25</sup>
- *Regression Testing* – The Testing Strategy included regression testing in the list of testing types, but failed to include a description or details within the body of the document. Regression testing helps to ensure that previous fixes were adequate and did not cause any further problems. Regression testing should be executed after flaws and defects have been corrected and a new version of the application has been released for testing.<sup>26</sup>

The Testing Strategy should have included the procedures for selection of test cases, including the depth and breadth across functionalities, and removal of obsolete test cases from the regression test suite. The Testing Strategy should have described the criteria for entrance and exit of regression testing, as well as how failed regression tests are handled. Regression testing should provide visibility into the stability of the system after code changes. CMP's failure to describe regression testing methodology created a risk that the project team would misunderstand the procedures, goals and objectives, and this could potentially preclude a complete or reliable test result.

***The Testing Strategy did not require development of test cases to cover all code path scenarios.*** An analysis of the test scenarios and test cases indicates that CMP's test plan

---

<sup>24</sup> Thomas, Jann. (2013). *Defect Driven Test Automation* [White Paper]. Retrieved from LeadingAgile: <https://www.leadingagile.com/2013/07/defect-driven-test-automation>.

<sup>25</sup> Watson, Matt. (2015). *Ultimate Software Deployment Checklist & Deployment Plan* [White Paper]. Retrieved August 20, 2019, from <https://stackify.com/ultimate-checklist-app-deployment-success>

<sup>26</sup> Sharma, Lakshay. (2016). *Software Testing*. Retrieved from ToolsQA: <https://www.toolsqa.com/software-testing/software-testing>.

included both positive and negative test cases.<sup>27</sup> Other types of test cases, including those for alternate, boundary, edge, error, and exception code paths were not documented in either the Testing Strategy or the list of test cases and test scenarios provided. Gaps in test coverage greatly increase the likelihood that defects will remain undetected until production use.

- Industry standards and best practices maintain that test cases should be created and categorized for all possible paths the user may follow when using the system.<sup>28</sup> By following this approach, testing best simulates real-world use by executing test cases that mimic real user journeys and critical business transactions through the application.
- When testers exercise the system in unusual and unexpected ways as would real-world users unfamiliar with the system, testers are more likely to encounter a higher number of use-based defects before the system is promoted to production.<sup>29</sup> Testing that does not cover both common and uncommon user interactions results in the risk that real-world users will experience interruptions, flaws and defects while using the system.<sup>30</sup> In order to balance quality, speed, and cost of testing a software application, organizations must employ a testing strategy that covers a wide range of code paths and diverse scenarios.

***The Testing Strategy did not include sufficient details on how test data is identified and selected to help ensure the test scenarios will exercise all code paths.*** Best practices agree that test coverage, completeness, and effectiveness depend mainly on the quality of test data, and as such recommend that software development efforts employ Test Data Management (TDM).<sup>31 32</sup>

- Using TDM, test data needs are analyzed for all possible test flow outcomes (e.g. positive, negative, alternate, error) and test conditions (e.g. boundary, edge). Using a sub-set of production or production-like data, single sets of test data are selected for each scenario. Repetitive test data for similar test cases are not required, thus reducing unnecessary execution time while maintaining or improving test quality and efficiency.

---

<sup>27</sup> TLCG-001-0185, Attachment 1; OPA-008-014.

<sup>28</sup> Jacobson, Ivar. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Essex, England: Addison-Wesley Professional.

<sup>29</sup> TryQA. (2008). *What is Use case testing in software testing?* Retrieved 20 August 2019, from <http://tryqa.com/what-is-use-case-testing-in-software-testing>.

<sup>30</sup> Devi, Indumathi. (2018). *Do More with Less in Software Testing* [White Paper]. Retrieved from Infosys: <https://www.infosys.com/IT-services/validation-solutions/white-papers/Documents/software-testing.pdf>.

<sup>31</sup> Bagare, Praveen and Desyatnikov, Ruslan. (2018). *Test Data Management in Software Testing Life Cycle - Business Need and Benefits in Functional, Performance, and Automation Testing* [White Paper]. Retrieved 20 August 2019, from Infosys: <https://www.infosys.com/it-services/validation-solutions/white-papers/documents/test-data-management-software.pdf>.

<sup>32</sup> Vivek Motwani. (2001). *The When & How of Test Automation* [White Paper]. Presented at the QAI 3rd Annual International Software Testing Conference, India, 2001.



Furthermore, conducting repetitive test data for similar test cases can be problematic in that it can skew test results.

- Without proper TDM, there is a risk that test data will not correctly and completely sample production data. Defects related to test data may remain undetected until the software is deployed in a production environment. Poor TDM may also result in data corruption or invalid test results (pass or fail) when multiple testing teams use the same environment.

The Testing Strategy developed by Deloitte and adopted by CMP for testing the SmartCare application did not follow best practices and lacked sufficient details of test methodology, scope, approach and goals to be an effective and useful document. Based on industry standards and best practices, the list of test types was incomplete; many of the test type descriptions, goals, and objectives were incorrectly or inadequately defined, the test case coverage was not discussed, and the selection, management, and intended use of test data was not described. These are critical components of a testing strategy that must be documented and communicated to the project team to help ensure the process is clear, actionable and maintainable and that the test phase of the SDLC can be successfully completed. This is particularly true with a project like SmartCare, which involved more than 240 resources, 32 vendors in 6 geographical locations and required interaction with 52 interfaces. The Testing Strategy as reviewed and approved with the stated flaws and omissions precluded the successful completion of testing of the SmartCare application. A qualified reviewer with experience in management of testing for a system of this size and complexity should have known from the minimal details provided in the Testing Strategy that it was insufficient and recommended rejection of this document.

***The testing conducted did not follow the approved Test Strategy.*** CMP did not follow testing best practices, nor did they conduct testing in accordance with their own approved but flawed Testing Strategy. Several changes were made to the testing approach but the Testing Strategy document was not updated.

At the technical conference on April 29, 2019, in the 2018-00194 docket, CMP employee Donna McNally, the management lead on the SmartCare project, testified that the testing approach was modified to include an additional integration test cycle (ITC 5) and to allow for test cycles to

overlap and run in parallel instead of serial or Waterfall sequence.<sup>33</sup> This required use of different environments for different test types (e.g. integration, performance, and batch).

In a Waterfall, or linear-sequential model, the testing for each test type must be completed before the next phase can begin; there is no overlapping in the phases.<sup>34</sup> Testing occurs as a sequence of stages in which the output from the completion of each test type or phase becomes the input for the next. Following this model and industry standards and best practices, system testing and user acceptance testing (UAT) must run one after another and never in parallel. UAT must be the final testing phase and must occur after all other testing is complete; this is true for all testing methodologies.<sup>35 36 37</sup> CMP's abandonment of the Waterfall approach to testing and their failure to replace it with any accepted testing methodology prevented management from validating that the system met their business requirements.

Integration Test Cycle 5 (ITC 5) was added to the SmartCare project in September 2017 and scheduled to run September to October, with a Go-Live date set for October 30, 2017. This fifth round of integration testing focused on the MDM interface, the other integrations, the business processes, and UAT, all scheduled to run during that same timeframe.<sup>38</sup> As such UAT was executed within that final cycle of integration testing<sup>39</sup>.

Running ITC 5 and UAT in parallel risked distributing a flawed and immature system to the business users for validation. This creates a potential for duplicated bugs introduced by business users and testers resulting in expanded turnaround time for defect analysis and resolution and slower development team response due to unanticipated load. There is also a risk that management will partially lose control over both releases and test environments, especially if there are separate test environments for ITC 5 and UAT. Final results of these risks

---

<sup>33</sup> Docket No. 2018-00194, Tr. 4/29/19 p. 17.

<sup>34</sup> Sharma, Lakshay (2017). *User Acceptance Testing – UAT*. Retrieved from ToolsQA: <https://www.toolsqa.com/software-testing/user-acceptance-testing-uat>.

<sup>35</sup> TestMatick. (2016). *Testing Software Services: Reasons to Avoid Overlapping UAT and System Testing*. Retrieved from TestMatick: <https://testmatick.com/testing-software-services-reasons-to-avoid-overlapping-uat-and-system-testing/>.

<sup>36</sup> ProfessionalQA. (2019). *Entry and Exit Criteria in Software Testing*. Retrieved from: <http://www.professionalqa.com/entry-and-exit-criteria>.

<sup>37</sup> The Institute of Electrical and Electronics Engineers. (1993). *IEEE Guide for Software Verification and Validation Plans* (IEEE Std 1059-1993).

<sup>38</sup> Docket No. 2018-00194, Tr. 4/29/19.

<sup>39</sup> Tr. 6/13/19.



are the inability to complete testing in time; thus, the product is not fully validated by the business users as meeting all business requirements and needs.

Changes of this magnitude should have prompted a formal change request that included analysis of impact to project baselines, assessment of risk and planned mitigations. This information should have been presented to project leadership to obtain formal approval of the change prior to it taking effect. It was not.

The Testing Strategy was described as a living document and stated that it would be updated “as information changes or becomes available.” (Section 1.1 Purpose, p. 4). This is considered a best practice. As a practical matter, given the number of teams impacted, the Testing Strategy document should have been updated to demonstrate that the change was authorized by CMP project leaders and to help ensure adequate communication concerning the testing that took place. Test managers from different teams and test types must be able to trust the program documents to reflect authorized change, the current approved testing guidelines and the plan of approach for achieving test objectives. Ms. McNally stated that there were no revisions made to the Testing Strategy document.<sup>40</sup> The 4/14/2016 version represented the last approved version of the Testing Strategy.

***Testing data did not support CMP’s approved criteria for Go-Live.*** The following criteria were specified on page 17 of the Testing Strategy:

- “All scenarios and custom development objects have been tested at least twice.”
- “All test exceptions have been re-tested as applicable and accepted.”
- “Zero open high or medium priority defects; unless there is a workaround.”

According to Ms. McNally someone at CMP made a “conscious decision” to change the criteria for Go-Live from “no high or medium priority” to “no critical or high” defects.<sup>41</sup> This represented a lowering of criteria for Go-Live. This is a substantive change to the Testing Strategy and CMP’s expectations of acceptable quality. The CMP project manager should have formally documented the proposed change as a change request. The impact of the proposed change should have been analyzed and the associated risk assessed. This information should have been provided to CMP leadership for formal review prior to a decision.

---

<sup>40</sup> Tr. 6/13/19 at 123.

<sup>41</sup> *Id.* at 148.

Neither the approved criteria nor the revised criteria were supported by program data at Go-Live. The test case execution results do not provide details on number of retests.<sup>42</sup> The list of valid open defects does not include workaround status for all critical, high or medium priority defects.<sup>43</sup>

Test execution results and defect status must be documented accurately and completely, and be accessible to allow the program team to determine if test exit criteria have been met and the software is ready for production deployment. The information provided regarding the number, priority ranking and availability of workarounds was not consistent or complete and did not demonstrate achievement of the Go-Live criteria approved by project leadership.

CMP's testing did not follow best practice. Test case coverage and test data management was not planned to validate all code paths. Testing did not demonstrate achievement of agreed upon criteria necessary for the system to Go-Live. CMP's approach and testing practices increased the risk that defects would remain undetected prior to (and after) Go-Live.

**Finding 3: CMP's defect management was inconsistent and did not provide information necessary for key decisions.**

The defect management process described in Section 9 at page 18 of the Testing Strategy document was incomplete and inconsistent with best practices and industry standards. CMP's responses to OPA-008 in this docket indicate that testing teams did not consistently adhere to the defect management process in the Testing Strategy, nor did they follow best practices.

***Defects were not consistently managed in a standard tool.*** Section 2.3 (Integration Testing, p.8) of the Testing Strategy, states: "HP Quality Center will be used for...managing defects." A best practice is that test teams across phases should share a defect management tool.<sup>44</sup> Defects discovered in unit and string testing phases in the development and implementation of SmartCare should have been managed in the HP Quality Center (HP QC) tool along with the defects discovered in later test phases. This would have allowed CMP to analyze, understand, and manage the quality throughout the software development life cycle by providing visibility into areas of instability uncovered in previous test phases, thus allowing for adjustments in test plans and scenarios.<sup>45</sup>

---

<sup>42</sup> OPA-008-017, Attachment 1.

<sup>43</sup> TLCG-001-031, Attachment 1.

<sup>44</sup> Eriksson, Ulf. (2011). *5 Steps That You Can Use As User Acceptance Testing Checklist*. Retrieved 2 February 2019, from ReQtest: <https://reqltest.com/testing-blog/5-steps-to-successful-acceptance-testing>.

<sup>45</sup> The Institute of Electrical and Electronics Engineers. (1993). *IEEE Guide for Software Verification and Validation Plans* (IEEE Std 1059-1993).

CMP deviated from best practices by not consistently using the HP QC tool. Rather, management used Excel spreadsheets managed in SharePoint to document test scenarios, test steps and defects for unit and string testing.

Without a common and accessible defect management tool for full disclosure of defect statistics and remediation results, there is a risk that the testing CMP should be doing today will not be performed in areas of historic defect density, and latent or unresolved defects will remain undetected.

**Defects were not consistently mapped to requirements.** Per Section 6 of the Testing Strategy (Requirements Traceability Process, p.22), requirements and test cases were to be loaded into the HP QC and “forward traceability” was supposed to be established between each requirement and the one or more test cases that covered the requirement. As such and as per best practices, defects detected during testing could be mapped to requirements through the failed test case.<sup>46</sup> Forward traceability is the mapping of each requirement to test scenarios to test cases and, ultimately, to defects. Forward traceability ensures that the evolution of the project design and implementation is following the proper direction and ultimately that the project team is building the right product.

In response to OPA-008-016, CMP stated: “Although testing defects are traceable back to requirements, not every defect recorded in HP ALM is a testing defect. Defects could be recorded as a result of other processes in implementation. Process examples include Dress Rehearsal and Parallel Bill Test.” This approach is contrary to best practice.

By definition, all defects are the result of a flawed, partial, or incorrect implementation of a requirement. The industry standard definition of a defect is an imperfection or deficiency in a work product where that work product does not meet its requirements or specifications and needs to be either repaired or replaced.<sup>47</sup> Defects identify flaws in a component or system that can cause the component or system to fail to perform its required function. When a defect is not associated with and traceable to a requirement, the root cause of the defect is more difficult to establish, the instability of the requirement will not be visible, and the identification or creation of test cases needed to retest or regression test the defect fix is more difficult.<sup>48</sup>

---

<sup>46</sup> Westfall, Linda. (2009). *Bidirectional Requirements Traceability*. Retrieved 2 February 2019, from The Westfall Team: <http://www.westfallteam.com/Papers/Bi-Directional%20Requirements%20Traceability%20updated%202009%2003%2018.pdf>.

<sup>47</sup> The Institute of Electrical and Electronics Engineers. (2009). *IEEE Standard Classification for Software Anomalies* (IEEE Std 1044 [2009 Edition]).

<sup>48</sup> Auadri, S.M.K & Farooq, Umar. (2010). Software Testing – Goals, Principles, and Limitations. *International Journal of Computer Applications* (0975 – 8887), 6(9), September 2010.

Defects that remain unlinked to a requirement are more difficult to prioritize. Requirements represent user expectations and as such, the impact of unlinked defects to user expectations cannot be reported.

CMP management did not consistently use a lifecycle management tool to capture defects across the SDLC test phases. Testers did not enter defects detected in unit and string testing into HP QC. This meant that during later test phases, testers and analysts could not easily compare new defects with previous defects to determine patterns or trends in quality. Such comparisons would have allowed CMP management visibility into the reintroduction of flaws reported in closed defects or the introduction of new defects concentrated in the same area of code as closed defects. This information could be useful to the development team when prioritizing defect fixes, or to the testing team when creating regression test cases. Without this information, there is a risk that there are unknown areas of the code that are historically brittle (i.e. easily broken) and need examination.

Adding defects from unit and string testing and mapping defects to requirements either directly or through test cases would have provided the ability to capture consistent defect metrics such as defect density, age, and failure rate.<sup>49</sup> Origins and root causes of areas of instability or unreliability in the application could have been analyzed using defect, test, and requirements data retained in HP QC. This analysis could have been used to develop a process improvement action plan to help improve system stability and quality, and help ensure defects were detected early in the test phases.<sup>50</sup>

CMP's failure to add tangible quality-centric artifacts (e.g., screen shots, error logs) and create relationships in HP QC impaired its ability to conduct reliable or complete root cause analyses (RCA). Performing RCA allows analysts to understand the full chain of events and relationships so that the actual root cause of system flaws is identified, fixed, and potential future defects can be prevented. RCA is a process that introduces organizational improvements and creates a lasting long-term effect on quality and reliability of software applications.<sup>51</sup>

---

<sup>49</sup> Software Testing Class. (2015). *Software Testing Best Practices – Into the Basics of Testing*. Retrieved 7 February 2018, from <http://www.softwaretestingclass.com/software-testing-best-practices-into-the-basics-testing>.

<sup>50</sup> Visitacion, Margo & Gualtieri, Mike. (2010). *Seven Pragmatic Practices To Improve Software Quality: A Forrester research report* (Forrester Research, Inc., 11 August 2010). Retrieved from: <http://dthomas-development.co.uk/wp-content/uploads/2014/05/forrester-research-report-seven-pragmatic-practices-to-improve-software-quality-on-the-web.pdf>.

<sup>51</sup> Lewallen, Raymond. (2008). *Benefits of Root Cause Analysis*. Retrieved from CodeBetter: <http://codebetter.com/raymondlewallen/2008/12/20/benefits-of-root-cause-analysis>.

***Failed test cases may not have resulted in a logged defect.*** Defects are identified through various types of testing including but not limited to scripted and unscripted, functional and non-functional, or manual and automated. Not every defect represents failure of a scripted test case. However, every failed test case should correspond to a logged defect.

In the SmartCare project, defects may or may not have been logged or linked to failed test cases. According to Ms. McNally, “Pass or fail is based on the user executing the test whether they pass or fail it, not necessarily if a defect is associated to it.” Reporting would reflect the failure, “if it was a fail, it was a fail, regardless of whether there was a defect or not. The user failed that test case for whatever reason.” Ms. McNally could not say whether there was a defect logged and linked to every failure as “it would depend on what the failure was.”<sup>52</sup> This violates standard practice for defect management and could have overstated the readiness for Go-Live.

The purpose of testing is to discover defects so they can be addressed. Defect lists are used to manage remediation of system problems. The fact that some defects may ultimately turn out to be duplicates or invalid or reflect a gap in requirements should not prevent testers from logging them. Marking a test case as “failed” communicates that something happened that didn’t align with an expected outcome. However, a defect record provides the details necessary to recreate, analyze, validate and prioritize the defect and then fix the problem. The practice of failing test cases without opening defects as described by Ms. McNally would prevent defects from being recognized, effectively managed, and fixed. This not best practice. I have not seen this done anywhere else. It undermines the whole intent of testing.

Go-Live criteria should have been based upon the number and type of defects that remained open, not a specific pass or failure rate for test cases. Using the practice described in the testimony above, the issue or condition that caused a tester to fail a test case would remain obscured and, without a defect record, it would not be considered in making the Go-Live decision.

**Finding 4: Traceability was not captured in a complete or structured manner.**

The current version of the PMBOK® (6<sup>th</sup> edition), p. 148, Section 5.2.3.2 describes the purpose of a Requirements Traceability Matrix (RTM) as follows:

“The requirements traceability matrix is a grid that links product requirements from their origin to the deliverables that satisfy them. The implementation of a requirements traceability matrix helps ensure that each requirement adds business value by linking it to the business and project objectives. It provides a means to track requirements throughout the project life cycle, helping to ensure that requirements approved in the requirements documentation are

---

<sup>52</sup> Docket No. 2018-194, Tr. 4/29/19, p. 48, lines 7-20.

delivered at the end of the project. Finally, it provides a structure for managing changes to product scope.”

The Testing Strategy document approved by CMP explains how CMP intended the project’s RTM to be developed and used:

“In the context of testing, requirement traceability represents the ability to verify that the solution delivered by the CMP CRM&B project has addressed all requirements. This is achieved through building linkages between individual requirements and the tests that are used to verify that the system has satisfied the requirements.

To permit traceability, each requirement must be uniquely and persistently labeled so that it can be referred unambiguously throughout the project. The Requirements are uniquely labeled in the blueprint deliverable – Requirements Traceability Matrix for CMP.”<sup>53</sup>

On this page and in the same section, the Testing Strategy described how the traceability would be managed and maintained in HPQC (aka HP ALM).

Per industry standards and best practices, an RTM should include requirements, the test cases that cover those requirements, and the defects detected during test case execution.<sup>54</sup> Following industry best practices, all defects detected during testing should be mapped to test cases, and all test cases mapped to requirements.<sup>55</sup> By enforcing this data integrity during testing and defect management, an RTM can be produced that demonstrates the traceability of defects to requirements and allows visibility into requirements that are flawed or incomplete.

In practice, however, the RTM approved by CMP management for the project included only requirements specific information.<sup>56</sup> Test case coverage was not provided in the RTM nor in the test cases and test scenarios matrix.<sup>57</sup>

The disposition of each requirement at Go-Live was not reported. Testimony provided at the June 13, 2019, technical conference confirmed that no single report, data extract or document

---

<sup>53</sup> OPA-008-011, Attachment 1, Section 6, p. 15.

<sup>54</sup> Miller, Steve. (2007). Test Coverage and Traceability. *Pragmatic Software Newsletter July 2007*. Retrieved from: [https://www.agileconnection.com/sites/default/files/article/file/2013/XUS27267853file1\\_0.pdf](https://www.agileconnection.com/sites/default/files/article/file/2013/XUS27267853file1_0.pdf).

<sup>55</sup> Simpson, John. (2009). Connect the Dots; Five Tips on Requirements Traceability. *BA Times*. Retrieved from: <https://www.batimes.com/articles/connect-the-dots-five-tips-on-requirements-traceability.html>.

<sup>56</sup> OPA-007-083, Attachment 2.

<sup>57</sup> TLCG-001-0185, Attachment 1.



was presented to decision makers to show that each requirement had been tested to plan, that all test cases for each requirement were executed, which had passed and which requirements still had outstanding defects. The absence of this type of traceability and reporting is contrary to the best practices cited above and does not align with the traceability included in the Testing Strategy approved by CMP.<sup>58</sup>

Industry research confirms that there is clear benefit in terms of higher software quality (as defined by number of defects per component) from recording and maintaining traceability information within a project.<sup>59 60</sup> Capturing traceability in an unstructured or incomplete manner will lead to reduced system quality, expensive iterations of defect corrections, incorrect prioritization of defects, and increased project costs.<sup>61 62</sup>

Relying on defect counts alone to measure implementation readiness is risky. When defects are appropriately logged, they provide visibility into known deficiencies. They do not allow for assessment of risk associated with unfound defects. Projects record defects for conditions they test. Project constraints can lead to decisions resulting in gaps in planned testing versus actual testing conducted. Traceability allows the project to assess the risk associated with these gaps and identify the functionality and likelihood of impact in order to develop appropriate mitigations and contingencies.

Without traceability, CMP did not know as of Go/No-Go which business requirements had been adequately tested, which had been delivered with defects and which had not.

#### **Finding 5: Risk management did not align with best practice or the approved plan.**

By definition, projects are unique undertakings with varying degrees of complexity, subject to constraints, assumptions and stakeholder expectations. The level of risk varies from project to

---

<sup>58</sup> 6/13/19 Tr. p. 156, lines 6 – p. 157 line 12.

<sup>59</sup> Rempel, Patrick & Mäder, Patrick. (2017). Preventing Defects: The Impact of Requirements Traceability Completeness on Software Quality. *IEEE Transactions on Software Engineering*, 43, pp. 777-797. doi: 10.1109/TSE.2016.2622264.

<sup>60</sup> Kpodjedo, S., Ricca, F., Galinier, P., Guéhéneuc, Y., Antoniol, G.. (2011). Design evolution metrics for defect prediction in object oriented systems. *Empirical Software Engineering*, 16(1), pp. 141–175. doi: 10.1007/s10664-010-9151-7.

<sup>61</sup> Li, W., Vaughn, R., & Saiedian, H. (2002). Pre-Requirements Traceability. In John J. Marciniak (Ed.), *Encyclopedia of Software Engineering* (Vol. 1, pp. 1155-1160). Retrieved from: [https://www.researchgate.net/publication/235353253\\_Pre-Requirements\\_Traceability](https://www.researchgate.net/publication/235353253_Pre-Requirements_Traceability).

<sup>62</sup> Packer, Dan. (2018). *The Importance of a Requirements Traceability Matrix*. Retrieved 15 February 2019, from Plutora: <https://www.plutora.com/blog/requirements-traceability-matrix>.

project but all projects have risks that must be managed. The PMBOK® (6<sup>th</sup> edition) devotes 70 pages to the subject and states that “the effectiveness of Project Risk Management is directly related to project success.” This is consistent with my own professional experience.

One of the primary objectives of risk management is to decrease the probability and/or impact of negative risk in order to optimize the chances of project success. It involves planning, risk identification, analysis, risk monitoring, risk response planning, and response implementation.

The Project Charter was approved by CMP.<sup>63</sup> It provides risk management guidance and the need and purpose of risk management for the project on pages 32-34.

***Risk tolerance and the project’s primary constraint were not explicitly defined.*** According to the Project Charter, setting strategic priorities for the project is the responsibility of the Executive Committee. It establishes the project manager’s responsibilities for the risk management methodology, inclusive of but not limited to:

- Setting standards for the risk management process
- Managing and controlling the risk management process

To manage risk effectively, the project team needs to know what level of risk exposure is acceptable in pursuit of project objectives, and specifically, what level of variation is acceptable for each objective. This information, often referred to as the project’s risk profile or risk tolerance, should be explicitly stated and is used to tailor project management processes and develop the definitions used to quantify risk and assess project deliverables.

In the course of managing projects, invariably hard decisions need to be made to keep things moving toward a successful completion. Quality is constrained by the project’s budget, schedule and scope (features). The project manager can make tradeoffs between constraints. Changes in one constraint (scope, schedule or cost) necessitate compensating changes in others or else quality will suffer. The project’s primary constraint is used to select acceptable tradeoffs between project scope, schedule and cost and should influence the tailoring of best practice and controls used to govern the project.

In the case of the CMP SmartCare implementation, when the project fell behind schedule, CMP could have prioritized or potentially reduced what the project must deliver (i.e., reduce scope). Management could have allocated more resources, increasing cost, to meet the scheduled Go-Live date. Or, they could have opted to push out the Go-Live date, extending project schedule

---

<sup>63</sup> OPA-008-001, Attachment 1.



duration. Which choice or combination of choices is best should have been determined based upon which one of the following statements is true—that is, which constraint is primary:

- Scope is primary. Cost and schedule are secondary. The project must deliver all product functionality, services, and results in the approved scope.
- Timing (i.e., schedule) is primary. Cost and scope are secondary. SmartCare must Go-Live on the date specified in the approved schedule.
- Cost is primary. Schedule and scope are secondary. The project must not exceed the approved budget amount.

At the technical conference on April 29, 2019, in the 2018-00194 case, Ms. McNally stated that quality was paramount. She stated that CMP wanted to “ensure the quality of the system” and that it was “focused on quality.”<sup>64</sup> Whether or not CMP’s primary constraint was scope, timing, or budget remains unclear. The planning documents approved by CMP did not provide this information. In response to OPA-007-008, the Company indicated that the required functionality (i.e., scope) was the primary constraint and that budget and schedule were adjusted as necessary:

“While all aspects of the SmartCare project were rigorously planned and managed, the most important element was delivery of a quality solution that met the business and customers’ required functionality. Budget and schedule parameters were adjusted as necessary in order to assure the critical solution functions were aggressively tested and delivered as designed.”

During that same technical conference, Mr. Stinneford’s testimony explained that costs influenced the modification to the approved testing approach. He stated:

“[T]he decision to adopt the test schedule and implementation schedule that was pursued was a decision that had to be -- at least take into consideration the cost implications. And although the quality of the delivered product was the paramount concern, that had to be balanced with the implications of extending the schedule because there is an expense to doing that. The project run rate was substantial at that point given the number of resources that were committed to the project. So we certainly pursued what we thought was a reasonable compromise that didn’t result in excessive project costs and ensure that the quality of the delivered project would not be compromised.”<sup>65</sup>

---

<sup>64</sup> Docket No. 2018-00194, Tr. 4/29/19 at 15]

<sup>65</sup> *Id.* at 22-23.

If quality was paramount and delivery of system functions (i.e. scope) was the primary constraint as suggested by Ms. McNally's testimony, quality could have been safeguarded by adding resources to complete all planned testing before the scheduled deadline and/or the schedule could have been modified. If cost was the primary driver of decision making, management could have helped to maintain quality by looking for opportunities to reduce the scope of what would be deployed at Go-Live. Instead, CMP management significantly modified its approach to testing, held to a fixed date without making adequate compensating changes to scope and/or cost and quality predictably suffered.

***The processes planned and adopted for project management did not provide the rigor necessary to effectively manage risk in a project of this size and complexity.*** The PMBOK® describes good practices that are applicable to most projects most of the time. However, PMBOK makes it clear that even standards approved by ANSI require that the sponsoring organization and project manager work together to tailor processes to the unique environment and characteristics of a specific project. It provides guidance on tailoring project management processes and how tools and techniques can be applied to projects.

The more complexity contained within the project, the more sophisticated the processes used for project management need to be to effectively manage risk. This project was highly complex. It was conducted by virtual, cross-functional, multicultural, geographically dispersed teams from different companies with disparate methods and tools. Robust, well-managed processes were essential to help ensure adherence to common expectations, project norms and standardized methods of measuring, monitoring, and reporting of project performance and product quality.

Listed below are four examples of project management processes that compromised CMP management's ability to consistently and aggressively manage risk:

- ***Risk Management*** - The Project Charter<sup>66</sup> describes the need and purpose of risk management. It acknowledges the need for "consistent monitoring and aggressive management" to limit project exposure and describes it as "an iterative process that will occur throughout the project life cycle."<sup>67</sup> In practice, CMP's risk management was neither consistent nor aggressive. A coordinated project-wide approach is needed to help ensure efficient and effective management of risk in a project of this profile. This standardization and rigor were absent in the risk management approach approved by CMP. The means of documenting risk varied throughout the project. Lapses in risk

---

<sup>66</sup> OPA-008-001, Attachment 1, pp. 32-34.

<sup>67</sup> *Id.* p. 32.

management practices were noted. At the June 13, 2019, technical conference Ms. McNally described risk management practices that deviated significantly from best practice.<sup>68</sup> CMP did not maintain a centralized master list nor did it establish a consistent method of managing risk. Given the critical nature of this project from the outset, the risk associated with compressed testing and the lack of prudent risk management was especially concerning.

- **Schedule Management** - The SmartCare project's schedule management processes did not call for critical path analysis on an integrated project schedule. A project's critical path is comprised of the sequence of activities, regardless of responsible entity, that represents the single longest path through a project and determines the project's shortest possible duration. Thus, any delay associated with any activity on the critical path jeopardizes the scheduled end date of the project. In a project schedule comprised of thousands of tasks, this analysis is critical to manage risk.

When asked if the project routinely used any form of critical path analysis to manage the schedule, the response was:

"There were a lot of critical paths in the project depending on where you were because you're delivering -- the system is more than a billing system. It provides a lot of business functions, and for each of those business functions, they had their own critical paths of delivery of those business functions. So those, once again, were assessed within the functional teams and then overall by the project lead team. You know, there were certain items that were definitely critical path to the overall project that were always dealt with at the project lead team and steering committee level, but within the individual business processes, there were also critical path items that were managed."<sup>69</sup>

CMP project management did not appear to know or understand the importance of establishing a single critical path for management of project risk, and it did not make it a priority to learn this. Tracking of "many critical paths" at the individual business process level would not yield the sequence of activities, regardless of responsible entity, that represents the single longest path through a project and determines the project's shortest possible duration. In a project schedule comprised of thousands of tasks, the impact of delays may not be accurately assessed without some form of critical path analysis. Thus, the ability to identify risks early when there is time to mitigate them and avoid impact to the project is significantly compromised.

---

<sup>68</sup> Tr. 6/13/19 at p. 157, line 21 – p.163 line 9.

<sup>69</sup> Docket No. 2018-00194, Tr. April 29, 2019 at p. 9 .

The Schedule Management plan did not call for loading of resources with applicable costs in an integrated project schedule. This basic functionality is available in schedule management tools such as Microsoft Project. While not commonly used for small projects, in a larger project such as the SmartCare implementation, it would have provided downstream impact of a specific delay and would provide early warning of risks. It could project when costs would outpace completion of scope based upon current status and burn rate. CMP's decision not to do this was not in conformance with best practices for a project the size and complexity of SmartCare.

The schedule management practices used on the SmartCare implementation compromised the project's ability to proactively monitor and aggressively manage risk.

- **Quality Management** – The level of detail in the quality management section of a program management plan is dictated by the risk profile and requirements of the project. At a minimum, the quality management section should describe how policies, procedures and guidelines will be implemented to achieve defined and documented quality objectives. It should contain quality standards applicable to the project, quality objectives, and quality roles and responsibilities. It should specify the deliverables and processes subject to quality review, quality control and management methods, quality tools and procedures for dealing with nonconformance, corrective actions and continuous improvement.

There is only a single paragraph in the quality management section of the SmartCare Program Management Office Plan and it is inadequate. This is the paragraph:

“Project quality standards, practices, resources, specifications, and the sequence of activities relevant to this project will be monitored. Responsibility of Management, Document Management and Control, Requirements Scope, Design Control, Development Control, Testing and Quality Assurance, Risks and Mitigation, Quality Audits, Defect Management, and Training requirements. As long as a project has defined objectives and deliverables, there should be a project quality plan to measure the delivery and process quality.”<sup>70</sup>

Seeking further information relevant to quality management, OPA-007-019 asked, “How was team performance monitored?” The response simply stated that project team members were accountable to meeting project goals and milestones. This response underscores the inadequacy of quality management processes established for the project.

---

<sup>70</sup> OPA 007-005 Attachment 1; TLCG 001-016 Attachment 1 at p. 10.

As noted on pages 64-65 of the Liberty Report, measures of quality were not represented in metrics including Executive Scorecards and Steering Committee Meeting reports. If, as stated in response to OPA-007-008, quality was the primary constraint for the project, CMP should have rigorously monitored quality processes and associated metrics should have been featured prominently in reporting.

- **Third-Party Management** - Given the number of vendors and interfacing entities, accountability would need to be reinforced through rigorous contract management and business relationship management processes. The governance model provided in response to OPA-007-005 (Figure 1: Governance Model) acknowledges a need for “Third-party management” but the Project Management Office Plan contains no section by that name. There is no indication in the Project Management Office Plan of the process that will be used for business relationship management with interfacing entities. It does contain a Contract management section comprised of only two sentences:



**Figure 1: Governance Model  
(OPA-007-005)**

“The System Integrator contract is a Fixed Bid agreement. Invoices will be reviewed for compliance against the bid and schedule of payments and any/all variable costs will be closely scrutinized.”

CMP’s failure to put these quality management requirements into writing in the Project Management Office Plan leaves quality management at risk of shifting expectations, demands and approaches of management, particularly if there are personnel changes in the manager positions.

**CMP’s ability to monitor, mitigate and respond to risks was compromised.** Assessing risk, establishing contingency plans and identifying trigger events and responsibilities for initiating mitigating actions are the responsibilities of the project manager according to the Project Charter.<sup>71</sup> The ability to detect risk early is critical to lessening the likelihood of risks becoming

<sup>71</sup> OPA-008-001, Attachment 1.

issues and to mitigate the impact of risks that do. This is exemplified by the level of risk response planning conducted for deployment of the SmartCare system as described in Finding 6.

**Finding 6: The risk response planning for Go-Live did not include an executable contingency or rollback plan.**

There are a number of ways to implement a new system. Three of the most common are:

- Parallel implementation – In this approach, the old and the new system are run in parallel, so that the system can be thoroughly validated in production and users can get used to the new system, and meanwhile do their work using the old system.
- Phased Implementation – In a phased implementation, the system is deployed incrementally with each deployment bringing the system closer to full implementation.
- Big Bang Implementation – In this approach the switch between using the old system and using the new system happens on one single date—the so-called instant changeover of the system. Everybody starts to use the new system at the same date and the old system will not be used from that moment on.

The SmartCare project utilized the Big Bang Implementation approach. Of the three approaches described above, it is riskiest because the system must be fully ready for use on day one and there are fewer learning opportunities incorporated in the approach. This type of implementation can be successful. However, it is prudent to thoroughly assess the risks associated with Go-Live and have well thought out, executable plans in the event risks are realized. Prudent risk management would call for at least two types of plans:

- Contingency plans for system implementation are designed to be used in the event an organization decides for whatever reason not to Go-Live with the new system. They should include specific measurable criteria or triggers that will prompt their use. They detail the activities that will need to be carried out in event the plan is invoked.
- A rollback plan is similar and is used when the sponsoring organization decides for whatever reason to reverse or “roll back” the implementation of a system after it has been deployed (i.e., post Go-Live). Like the contingency plan, it should include specific, measureable criteria that serve as triggers for invoking the plan as well as detailed activities that will need to be carried out if rollback is determined to be the best course of action.

If a contingency or rollback plan is invoked, it is likely the situation is already fraught with risk. In this environment, a well thought out plan that includes roles, responsibilities, and very detailed activities is essential to minimize unnecessary disruption. Such a plan minimizes or eliminates last-minute, rushed decision-making, when errors in judgment are more common.

The SmartCare project had an overly broad contingency and roll back strategy. CMP failed to develop a well thought out executable plan detailing what specific steps would be involved or pre-defined, objective metrics to assess when each plan should be used.

During the April 29, 2019, technical conference for Docket 2018-00194 this point was discussed at some length. When asked if reverting back to the legacy system would have been impossible after cutover was deemed complete, Ms. McNally stated, “I don't know if reversal would have been impossible. It would have been time consuming and we would have had to raise – weigh the risks and impacts of trying to go back as opposed to going forward if something would have happened.” Mr. Stinneford concurred saying, “I would say it was impractical but probably not impossible. You know, particularly the disruption that that would have created may have been worse in terms of, you know, trying to deal with the storm and everything else. If we had tried to reverse course at that point in time, it probably would have been even more disruptive.”<sup>72</sup>

The danger of not having an executable contingency plan is that the pressure to meet the scheduled Go-Live mounts as the target date approaches. Project budgets are often dwindling or overdrawn and costs of delay may be a factor. A Go-Live date may have been announced and stakeholder expectations are a consideration. Projects involving multiple vendors and interfacing business partners must consider the impact to those organizations. With so much time and money already spent, invoking even a fully executable contingency plan is often perceived as failure as opposed to the prudent course of action. When no such detailed plan exists and predefined triggers have not been agreed upon, as was the case with the SmartCare project, the likelihood that the contingency option will be used, even in face of mounting risk, is even less.

Without a viable rollback plan, project managers and others involved in the effort will feel pressure and do things in an emergency mindset “to fix the system” that they would not ordinarily do. Examples I have personally observed on other projects include such things as suppressing audit functionality and suppressing edits to improve performance, making changes to the production system or data outside of normal controls and without adequate testing to keep transactions flowing, implementing invalidated workarounds, and cutting corners in ways that can make the situation worse.

---

<sup>72</sup> Docket No. 2018-194, Tr. 4/29/19, p. 49-50.



### 3 Conclusions

Project practices employed by CMP in the implementation of SmartCare significantly deviated from those that would be considered prudent for a project of this size and complexity.

- Practices used for traceability, testing, defect management, and reporting during the implementation of SmartCare did not objectively demonstrate the readiness of the system or fully assess the prevalence of defects at the time that CMP made the decision to Go-Live with the SmartCare system.
- Neither the Testing Strategy approved by CMP nor the testing practices ultimately adopted by CMP aligned with best practices. Test planning did not acknowledge the need for test cases for alternate, boundary, edge, error, and exception code paths. Test cases provided did not provide this coverage, increasing the likelihood that a significant number defects would remain detected.
- The defect management practices used by CMP did not ensure all defects would be documented, effectively managed or accounted for in decision making.
- The risk management processes adopted and risk response planning performed by CMP did not align with best practice.
- CMP did not have an executable contingency plan for Go-Live nor did they have an executable rollback plan. This would have increased pressure to Go-Live and to continued use of the new system regardless of quality and known risks.

These deviations by CMP increased project risk and the likelihood that defects would remain undetected prior to implementation of SmartCare into production. CMP's failure to ensure traceability made it unclear which business requirements had been adequately tested, which had been delivered without defects and which had not.

Responses to questions and testimony provided in technical conferences as recently as June 2019 describing incident management, impact analysis, and defect management post Go-Live do not give any indication that practices in these areas have been improved.<sup>73</sup> This does not provide confidence, even now, that testing is adequate or that all defects are currently being documented, effectively managed, reported or accounted for in decision making.

---

<sup>73</sup> Tr. 6/13/19, p. 163, line 10 – p. 180 line 9



## 4 Recommendations

### 1. **Conduct Validation Testing Under Supervision by Independent Third Party -**

Practices used for traceability, testing, defect management, and reporting during the implementation of SmartCare make it impossible to go back and objectively ascertain the readiness of the system or prevalence of defects at the time the system was implemented.

In order to validate that the solution currently in production is compliant with requirements, and to restore stakeholder confidence, CMP should be required to test the system as it should have been tested prior to Go-Live in accordance with applicable best practices.

Based upon the documentation and testimony provided, it is not clear if CMP leadership and staff possess the knowledge of best practices and have experience to apply them. It is possible that some or all of these best practices were purposely ignored. Furthermore, customer experience and the overall public reaction following CMP's implementation of SmartCare have undermined public confidence. Therefore, it would be advisable to have an independent third party selected by the MPUC verify that the plan for this validation testing effort aligns with best practice, confirm that the effort was executed to plan, validate that the results are complete and accurate and issue a report.

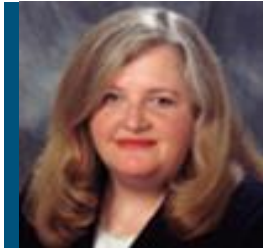
2. **Continuing Impact Analysis** - The defects discovered during this validation testing effort should be analyzed to identify transactions that may have been impacted during the period prior to the defect being fixed. Interim measures, such as temporary workarounds, should be identified to prevent further impact while the system is remediated. Transactions should be reprocessed and compensating adjustments made to correct incorrect billing.
3. **Payment of Costs** – Costs of the validation testing (as described above) and associated defect management, defect remediation, and any interim workarounds should be disallowed and/or paid for by the utility's shareholders. Until validation testing can be completed, an amount equivalent to the costs of pre-Go-Live test planning, test execution and defect management should be taken out of rates since these amounts, at least, were imprudently incurred, as shown in this testimony. The outcome of validation testing should be used to calculate a final portion of SmartCare costs associated with pre-Go-Live test planning, test execution and defect management that should be subject to disallowance.
4. **Action Plan to Restore Public Confidence** - Finally, CMP should develop an action plan subject to review by the MPUC. Once approved, it should be published to restore stakeholder confidence. The plan developed by CMP should be informed by the validation testing effort but, at minimum, should demonstrate a commitment to

---

consistently use best practices to implement, maintain and manage technology used to serve its customers.

The execution of this plan as well as the processes and practices used by CMP should be monitored by an independent third party who periodically reports directly to the MPUC for a period to be determined by the MPUC.

## Attachment 1: Laurel Arnold Resume



### Laurel Arnold, Senior Manager, BerryDunn

I have been helping organizations achieve their strategic goals and compliance objectives for more than 25 years. My experience includes strategic planning, project portfolio management, program and project management, quality assurance, independent testing, system implementation, business continuity, and disaster recovery planning.

Long standing client relationships have afforded me the opportunity to see strategic initiatives through from inception to implementation. I have been repeatedly called upon to objectively assess and provide recommendations for at risk projects and programs.

### Relevant Experience

#### **Independent Assessments, Independent Verification & Validation (IV&V) and Quality Assurance/Quality Control Services**

##### **Massachusetts HIX/IES Entities – IV&V Services (10/2012 to present)**

I lead the BerryDunn team providing the Massachusetts Executive Offices of Health and Human Services and the Commonwealth Connector Authority with Independent Verification & Validation (IV&V) services for implementation of the Massachusetts Health Insurance Exchange and Integrated Eligibility System. The IV&V team reviews both the project management and system delivery aspects of the project as conducted by the Commonwealth and its contracted vendor. We conduct deliverable reviews; verification and validation of automated code review and continuous integration results; cost allocation and financial status reports; review of expected and delivered reusability; independent assessment of implementation readiness; and issue and risk management.

**North Carolina Office of the State Auditor** – I led an independent audit of the State IT Services Enterprise Project Management Office (EPMO) practices across a portfolio of projects to determine whether they were fulfilling their legislative charter to minimize project risk and maximize the likelihood of project success. (2007)

**Maine Bureau of Motor Vehicles (BMV) on behalf of Secretary of State** – I was part of a team responsible for an independent assessment and recommendations for a program of at-risk licensing and registration systems. (2004-2005)

**Maine Inland Fisheries and Wildlife on behalf of State Chief Information Officer** – I was part of a team responsible for providing an independent assessment and recommendations for recovery of the implementation of Maine sports licensing system (aka MOSES) project. (2003)

**Illinois State Board of Education, Office of Auditor** – I was part of a team that provided an independent assessment and recommendations for a program of at-risk student information system and grant system projects. (2001-2002)

---

## **Project Management and Quality Assurance Experience**

**State of West Virginia** - The following project management and quality assurance projects were conducted on behalf of the West Virginia Bureau for Medical Services (BMS) – (2003-2012)

- **Data Warehouse / Decision Support System (DW/DSS) Project Management**  
I provided project management oversight for the Bureau's Data Warehouse/Decision Support System re-procurement, including development of the Advance Planning Document (APD) and Request for Proposal (RFP), and assistance with the vendor selection process. She provided project management oversight for the successful implementation of the DW/DSS. (2011-2012)
- **PPACA Planning, Analysis, and Implementation Support**  
I led a team that helped the Bureau navigate the requirements of the Affordable Care Act. The purpose of this project was to determine the impact of this legislation on the State of West Virginia's Medicaid policy, processes, budget, and communication, and plan the initiatives needed to become compliant with the applicable provisions of the healthcare reform law. (2011-2012)
- **Project Management of MMIS Procurement, Design, Development, Implementation, and Certification**  
I led the BerryDunn team engaged to conduct West Virginia's first MITA State Self-Assessment, analyze the current Medicaid Management Information System and Fiscal Agent operations, develop an APD and RFP for the MMIS re-procurement, and provide project management through the procurement and implementation. (2008-2012)
- **State Medicaid Health IT Planning and Health Care Reform Consulting**  
I served as project manager for the development of West Virginia's Planning-APD requesting 90% Federal Financial Participation for a State Medicaid Health IT Plan (SMHP) under the American Recovery and Reinvestment Act (ARRA). Development of the P-APD involved development of an organization and governance structure for the State's Medicaid Health Information Technology planning, facilitation of stakeholder work sessions, development of a survey plan, deployment of a web-based survey, analysis and reporting of survey results, and development of the P-APD inclusive of work plan and budget for submission to the Centers for Medicare and Medicaid Services. The State received the funding requested, and I served as the project manager for the subsequent Statewide Medicaid Health IT Planning Initiative, which included development of a HIT Roadmap and project portfolio, procurement planning, and an Implementation APD. (2010-2012)
- **Project Management Support and QA Oversight of Pharmacy Point of Sale (POS) Implementation Recovery**  
Shortly after the initial implementation of its Pharmacy POS system, West Virginia invoked its rollback contingency plan, reverting to its prior system for processing of

pharmacy claims. I provided project oversight for the project recovery and system remediation. I worked with the State and Fiscal Agent teams to rebuild team morale, regain stakeholder confidence, identify the root causes of the initial failure, validate remediation of defects and objectively, verify readiness for re-implementation. (2003-2008)

- **QA Oversight of MMIS Implementation**

As lead QA monitor for West Virginia's MMIS Implementation, I worked with the State and the implementation vendor to integrate quality into project planning and processes and assess deliverables and services to ensure they were fit for their intended purpose and met stakeholder expectations. I served as a liaison between the State and vendor; facilitated risk mitigation, issue and conflict resolution; helped to broker consensus on acceptance criteria for testing, and implementation; and monitored contract compliance to ensure the State would not pay more than the total fixed cost specified in the Fiscal Agent contract for development, design and implementation activities. I subsequently served as the State's project manager for the federal certification the MMIS implementation. (2003-2008)

- **Portfolio Governance and Project Management Office**

I led the BerryDunn team, tasked with the development of a Project Management Office for the Medicaid program. I helped to establish a portfolio governance structure and board to manage the state's Medicaid projects. I worked with leadership to establish a framework and planning approach to implement the PMO. The PMO is responsible for ensuring that project management standards and protocols are implemented on all Bureau projects. (2008-2012)

### **System Development Experience**

#### **New Hampshire Department of Health and Human Services (DHHS) – *Enterprise Data Warehouse Project (2001-2002)***

I worked as part of a team that developed requirements for an Enterprise Data Warehouse. This included the development of business and technical recommendations for 24 user-defined reports; leading joint application design work sessions; and analyzing system-wide data models, data dictionaries, and system specification documents. I provided analysis, recommendations and specification for consolidation and enhancement of reports to better serve the information needs of stakeholder agencies.

#### **State Farm Insurance (1997 to 2000)**

I began my information technology career at State Farm Insurance at their corporate headquarters in Bloomington, Illinois as a system analyst for client systems.

As a member of the business continuity program, I led a team of specialists and writers that developed 1100+ disaster recovery plans covering enterprise operations throughout the US and Canada.

---

I completed project management training and provided project planning for enterprise system projects.

### **Education and Certifications**

BS, Computer Information Systems, Illinois State University

Certified Project Management Professional® (PMP®), Project Management Institute® (PMI®)

Prosci® Certified Change Management Practitioner (CCP)